There is claimed:

1. A method for recognizing a pattern in a design of an integrated circuit (IC), comprising:

a) identifying a pattern correspondence element in a pattern instance;

b) building a pattern tree corresponding to the pattern instance;

c) defining a list of candidate design correspondence elements in a design instance of the IC; and

d) iteratively, for each design correspondence element in said list of candidate design correspondence elements, comparing each rank in a tree representation of said design instance built around said each design correspondence element with corresponding rank in said pattern tree.

2. The method of claim 1, wherein the pattern instance is in an RTL description.

3. The method of claim 1, wherein the design instance is in an RTL description.

4. The method of claim 1, wherein the design instance and the pattern instance are created using a same synthesis engine.

5. The method of claim 1, wherein the comparing is done only using a topological comparison of the design instance and the pattern instance.

6. The method of claim 1, wherein rare design correspondence elements are chosen for comparison at a higher priority.

7. The method of claim 1, wherein design correspondence elements that are more likely to provide a successful comparison are chosen for comparison at a higher priority.

8. The method of claim 1, wherein said pattern correspondence element is an instance chosen as a starting point for matching between said given pattern instance and said design instance.

9. The method of claim 1, wherein said pattern tree comprises successive lists of fanin terminals, fanout terminals and pórts.

10. The method of claim 9, wherein the root of said pattern tree is the pattern correspondence element.

11. The method of claim 1, wherein the pattern tree is built using a sub-process comprising:

a) iteratively, for each node starting from said pattern correspondence element building a child node list;

b) for each node in said child node list, setting a rank level of said each node; and

c) sorting said child node list.

12. The method of claim 11, wherein a parent node is attached to said child node list of said parent node.

13. The method of claim 12, wherein said child node list comprises a list of fanin terminals.

14. The method of claim 11, wherein said fanin terminals comprise a list of input terminals.

15. The method of claim 12, wherein said child node list comprises a list of fanout terminals.

16. The method of claim 15, wherein said fanout terminals comprise a list of output terminals.

17. The method of claim 11, wherein said rank level is a number of terminal steps from said pattern correspondence element to a node having the rank.

18. The method of claim 17, wherein said terminals include at least one of fanin terminals and fanout terminals.

19. The method of claim 11, wherein said rank level is set to a rank level of a parent node of said child node list incremented by one.

20. The method of claim 19, wherein said rank level of the root of said pattern tree is set to zero.

21. The method of claim 19, wherein said sorting comprises a lexical sorting.

22. The method of claim 7, wherein the comparing is performed using a sub-process comprising:

a) selecting a rank from the tree representation of said design instance to be matched;

b) iteratively, for each node with the selected rank building a child node list;

c) sorting said child node list;

d) building a list of pattern nodes with the selected rank;

e) matching between said list of pattern nodes and said child node list; and,

f) repeating said steps a) through e) for all ranks in said design instance.

23. The method of claim 22, wherein said child node list comprises terminals of said design instance.

24. The method of claim 22, wherein a parent node is attached to said child node list of said parent node.

25. The method of claim 22, wherein permutable terminals are assigned equivalent node names.

26. The method of claim 25, wherein said child node list comprises a list of fanin terminals of said parent node.

27. The method of claim 35, wherein said child node list comprises a list of fanout terminals of said parent node.

28. The method of claim 22, wherein said sorting comprises a lexical sorting.

29. The method of claim 23, wherein when comparing nodes in said list of pattern nodes are matched against nodes in said child node list according to said nodes' order in said list of pattern nodes and said child node list.

30. The method of claim 29, wherein said nodes are considered matched if said nodes have the same identification name.

31. The method of claim 30, wherein said identification name comprises at least a terminal name.

32. The method of claim 30, wherein a report indicating failure is generated of nodes are mismatched.

33. The method of claim 1, wherein one of a computer aided design (CAD) system, a CAD program and a clock synchronization analysis tool is used to implement the process.

34. A computer program product, including computer-readable media with instructions to enable a computer to implement a process for recognizing a pattern in a design of an integrated circuit (IC), comprising:

a) identifying a pattern correspondence element in a pattern instance;

b) building a pattern tree corresponding to the pattern instance;

c) defining a list of candidate design correspondence elements in a design instance of the IC; and

d) iteratively, for each design correspondence element in said list of candidate design correspondence elements, comparing each rank in a tree representation of said design instance built around said each design correspondence element with corresponding rank in said pattern tree.

35. The computer program product of claim 34, wherein the pattern instance is in an RTL description.

36. The computer program product of claim 34, wherein the design instance is in an RTL description.

37. The computer program product of claim 34, wherein the design instance and the pattern instance are created using a same synthesis engine.

38. The computer program product of claim 34, wherein the comparing is done only using a topological comparison of the design instance and the pattern instance.

39. The computer program product of claim 34, wherein rare design correspondence elements are chosen for comparison at a higher priority.

40. The computer program product of claim 34, wherein design correspondence elements that are more likely to provide a successful comparison are chosen for comparison at a higher priority.

41. The computer program product of claim 34, wherein said pattern correspondence element is an instance chosen as a starting point for matching between said given pattern instance and said design instance.

42. The computer program product of claim 34, wherein said pattern tree comprises successive lists of fanin terminals, one fanout terminals and ports.

43. The computer program product of claim 43, wherein the root of said pattern tree is the correspondence element.

44. The computer program product of claim 34, wherein the pattern tree is built using a sub-process comprising:

a) iteratively, for each node starting from said pattern correspondence element building a child node list;

b) for each node in said child node list, setting a rank level of said each node; and

c) sorting said child node list.

45. The computer program product of claim 44, wherein a parent node is attached to said child node list of said parent node.

46. The computer program product of claim 45, wherein said child node list comprises a list of fanin terminals.

47. The computer program product of claim 44, wherein said fanin terminals comprise a list of input terminals.

48. The computer program product of claim 45, wherein said child node list comprises a list of fanout terminals.

49. The computer program product of claim 48, wherein said fanout terminals comprise a list of output terminals.

50. The computer program product of claim 44, wherein said rank level is a number of terminal steps from said pattern correspondence element to a node having the rank.

51. The computer program product of claim 50, wherein said terminals include at least one of fanin terminals and fanout terminals.

52. The computer program product of claim 44, wherein said rank level is set to a rank level of a parent node of said child node list incremented by one.

53. The computer program product of claim 54, wherein said rank level of the root of said pattern tree is set to zero.

54. The computer program product of claim 52, wherein said sorting comprises a lexical sorting.

55. The computer program product of claim 40, wherein the comparing is performed using a sub-process comprising:

a) selecting a rank from the tree representation of said design instance to be matched;

b) iteratively, for each node with the selected rank building a child node list;

c) sorting said child node list;

d) building a list of pattern nodes with the selected rank;

e) matching between said list of pattern nodes and said child node list; and,

f) repeating said steps a) through e) for all ranks in said design instance.

56. The computer program product of claim 55, wherein said child node list comprises terminals of said design instance.

57. The computer program product of claim 55, wherein a parent node is attached to said child node list of said parent node.

58. The computer program product of claim 55, wherein permutable terminals are assigned equivalent node names.

59. The computer program product of claim 58, wherein said child node list comprises a list of fanin terminals of said parent node.

60. The computer program product of claim 59, wherein said child node list comprises a list of fanout terminals of said parent node.

61. The computer program product of claim 55, wherein said sorting comprises a lexical sorting.

62. The computer program product of claim 56, wherein when comparing nodes in said list of pattern nodes are matched against nodes in said child node list according to said nodes' order in said list of pattern nodes and said child node list.

63. The computer program product of claim 62, wherein said nodes are considered matched if said nodes have the same identification name.

64. The computer program product of claim 63, wherein said identification name comprises at least a terminal name.

65. The computer program product of claim 63, wherein a report indicating failure is generated of nodes are mismatched.

66. The computer program product of claim 34, wherein one of a computer aided design (CAD) system, a CAD program and a clock synchronization analysis tool is used to implement the process.

67. A system for recognizing a pattern in a design of an integrated circuit (IC), comprising

a compiler adapted to generate a pattern instance and a design instance;

a correspondence element identifier adapted to identify correspondence element in the pattern instance and the design instance;

a tree generator to generate a pattern tree and a tree representing the design instance around the correspondence element; and

a comparison unit adapted to iteratively compare rank in the tree representation of said design instance corresponding rank in said pattern tree.

68. The system of claim 67, wherein the pattern instance is in an RTL description.

69. The system of claim 67, wherein the design instance is in an RTL description.

70. The system of claim 67, wherein the compiler is designed to convert the RTL description into synthesized representation.

71. The system of claim 67, wherein the comparing is done only using a topological comparison of the design instance and the pattern instance.

72. The system of claim 67, wherein rare design correspondence elements are chosen for comparison at a higher priority.

73. The system of claim 67, wherein design correspondence elements that are more likely to provide a successful comparison are chosen for comparison at a higher priority.

74. The system of claim 67, wherein said pattern correspondence element is an instance chosen as a starting point for matching between said given pattern instance and said design instance.

75. The system of claim 67, wherein said pattern tree comprises successive lists of fanin terminals, one fanout terminals and ports.

76. The system of claim 75, wherein the root of said pattern tree is the correspondence element.